

**RANCANG BANGUN APLIKASI PENGONTROL SISTEM
PENYIRAM TANAMAN BERBASIS *ARDUINO* DAN *ANDROID***



**Disusun sebagai salah satu syarat memperoleh Gelar Strata I
pada Jurusan Informatika Fakultas Komunikasi dan Informatika**

**Oleh:
AYASHA NINDA MAHARANI
L200170143**

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2021**

HALAMAN PERSETUJUAN

**RANCANG BANGUN APLIKASI PENGONTROL SISTEM
PENYIRAM TANAMAN BERBASIS *ARDUINO* DAN *ANDROID***

PUBLIKASI ILMIAH

oleh:

AYASHA NINDA MAHARANI

L200170143

Persetujuan untuk diujikan oleh:

Dosen Pembimbing



Ir. Bana Handaga, M.T., Ph.D.

NIK.793

28 Juni 21

HALAMAN PENGESAHAN

**RANCANG BANGUN APLIKASI PENGONTROL SISTEM
PENYIRAM TANAMAN BERBASIS *ARDUINO* DAN *ANDROID***

OLEH

AYASHA NINDA MAHARANI

L200170143

Telah dipertahankan di depan Dewan Penguji

Pada hari Kamis, 15 Juli 2021

dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. **Ir. Bana Handaga, M.T., Ph.D.**
(Ketua Dewan Penguji)
2. **Endah Sudarmilah, Dr. M.Eng.**
(Anggota I Dewan Penguji)
3. **Dr.Eng. Yusuf Sulisty Nugroho, S.T., M.Eng.**
(Anggota II Dewan Penguji)

(.....)

Anda

(.....)

Dekan
Fakultas Komunikasi dan Informatika



Nugroho, S.T., M.Sc., Ph.D.

NIK.881

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam publikasi ilmiah ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 15 Juli 2021

Penulis



AYASHA NINDA MAHARANI

L200170143

RANCANG BANGUN APLIKASI PENGONTROL SISTEM PENYIRAM TANAMAN BERBASIS ARDUINO DAN ANDROID

Abstrak

Keberadaan tanaman sebagai makhluk hidup sangat penting bagi kehidupan manusia. Tanaman bertahan hidup dengan memanfaatkan air. Proses penyiraman tanaman yang tidak terkontrol dapat berakibat buruk pada kelangsungan hidupnya. Serta proses penyiraman yang dilakukan secara manual sangat tidak efisien karena pemilik harus berada dalam satu tempat dengan tanaman miliknya. Tujuan dari penelitian ini adalah untuk merancang aplikasi pengontrol sistem penyiraman tanaman otomatis yang berbasis android dan arduino supaya dapat digunakan secara mobilitas. Aplikasi dirancang dengan metode SDLC waterfall menggunakan toolkit milik Flutter dengan bahasa perograman dart. Sedangkan perancangan sistem menggunakan software Arduino IDE. Hasil dari penelitian ini berupa aplikasi android yang dapat memonitor nilai kelembapan tanah pada tanaman dalam satuan waktu dan tombol on/off yang dapat menghidupkan dan mematikan pompa air penyiram tanaman. Pengguna dapat melihat nilai kelembapan tanah secara realtime melalui aplikasi android dengan menggunakan koneksi internet. Penggunaan ini menggunakan server firebase sebagai database dengan memanfaatkan fitur realtime firebase database milik server firebase. Sistem dan aplikasi yang telah dirancang diuji kegunaannya dengan pengujian terakhir aplikasi menggunakan usability testing. Hasil dari survey pada usability testing mendapatkan hasil bahwa aplikasi berfungsi sebagaimana semestinya dan mendapat kategori “Sangat Baik” sebesar 94,44% dari responden.

Kata Kunci: Android, Arduino, Firebase, Flutter

Abstract

The existence of plants as living things is very important for human life. Plants survive by utilizing water. The process of watering plants that are not controlled can have a negative impact on their survival. And the watering process that is done manually is very inefficient because the owner must be in the same place with his plants. The purpose of this research is to design an application for controlling an automatic plant watering system based on Android and Arduino so that it can be used for mobility. The application is designed using the SDLC waterfall method using Flutter's toolkit with the darts programming language. While the system design uses Arduino IDE software. The results of this study are an android application that can monitor the value of soil moisture in plants in units of time and an on/off button that can turn on and off the water pump for watering plants. Users can see the value of soil moisture in real time through the android application using an internet connection. This usage uses the firebase server as a database by utilizing the firebase server's realtime firebase database feature. The systems and applications that have been designed are tested for usability with the final test of the application using usability testing. The results of the survey on usability testing found that the application was functioning as it should and got the "Very Good" category of 94.44% of the respondents.

Keyword: Android, Arduino, Firebase, Flutter

1 PENDAHULUAN

Manusia bukan merupakan satu-satunya makhluk hidup yang ada di bumi. Manusia hidup di bumi berdampingan dengan makhluk hidup lainnya, salah satunya ialah tanaman. Keberadaan tanaman sebagai makhluk hidup sangat penting bagi kehidupan manusia. Tanaman memiliki banyak manfaat yang berguna untuk manusia. Untuk mendapatkan manfaat tersebut, tanaman harus tetap hidup tumbuh. Faktor penunjang tumbuh dan berkembangnya tanaman ialah pada proses penyiraman (Azzaky & Widianoro, 2021).

Tanaman bergantung kepada air untuk dapat bertahan hidup (Sreeram et al., 2018). Air dari dalam tanah akan diserap oleh tanaman sesuai jumlah yang dibutuhkan. Kandungan air dalam tanah yang sangat kering dapat menyebabkan tanaman menjadi layu, sedangkan jika kandungan air dalam tanah sangat berlebihan dapat menyebabkan turunnya kadar oksigen dalam tanah (F. Hidayat, 2019). Kedua hal tersebut dapat berakibat buruk pada kondisi tanaman yang dapat menyebabkan tanaman menjadi mati.

Kurang optimalnya perawatan tanaman tersebut dapat disebabkan oleh faktor waktu dan tenaga bagi pemilik tanaman (F. Hidayat, 2019). Fleksibilitas waktu yang dimiliki pemilik tanaman dan sifat dasar manusia yang mudah lupa berdampak pada kelangsungan hidup tanaman.

Proses penyiraman tanaman pada saat ini masih dilakukan secara manual dengan menyalakan pompa air secara manual maupun dengan mengambil air dari sumur yang masih membutuhkan waktu dan tenaga pemilik tanaman (Satria, 2019). Cara tersebut sangat tidak efisien karena pemilik tanaman harus mengontrol secara langsung di taman atau kebun. Kebiasaan mengontrol tanaman secara manual ini dapat diatasi dengan teknologi yang ada. Era globalisasi menuntut kita untuk hidup berdampingan dengan teknologi. Seluruh kegiatan manusia hampir tidak pernah luput dari teknologi (Ridarmin & Pertiwi, 2018). Kemajuan dan perkembangan teknologi menciptakan ponsel pintar yang biasa disebut sebagai *Smartphone*. *Smartphone* sendiri menggunakan sistem operasi yang berbeda-beda. Namun sistem operasi yang sedang banyak digunakan adalah sistem operasi milik *google* yaitu *Android* (Warangkiran et al., 2014). Saat ini hampir seluruh dunia menggunakan *smartphone* untuk beraktifitas (Satria, 2019).

Sebelumnya telah dilakukan penelitian oleh (Satria, 2019) dengan membuat Sistem Penyiram Perkebunan Berbasis *Android* dan *Arduino* Uno untuk menjalankan

sistem. Peneliti menggunakan Modul GSM untuk mengirimkan SMS sebagai perintah menyalakan dan mematikan pompa ketika sedang berada jauh dari tanaman.

Selain itu (F. Hidayat, 2019) melakukan penelitian dengan membuat Alat Penyiram Tanaman Otomatis menggunakan Sensor Kelembapan Tanah dengan Notifikasi *Whatsapp*. Pengujian yang dilakukan peneliti terdiri dari pengujian sensor soil moisture, pengujian lampu led, pengujian lcd, pengujian *relay*, pengujian pompa air dan pengujian *whatsapp*. Hasil pengujian didapatkan apabila kelembapan tanah yang didapat sensor soil moisture melebihi 6.5 maka pompa air akan menyala secara otomatis. Peneliti akan mendapat notifikasi melalui *whatsapp* setiap proses penyiraman dan memperoleh informasi kelembapan tanah melalui lcd.

Sistem Penyiram Tanaman Pintar Menggunakan Smartphone dan *Mikrokontroler Arduino* Berbasis Internet of Thing juga telah dibuat (Pambudi et al., 2020). Sistem tersebut dirancang peneliti menggunakan aplikasi smartphone *Blynk* yang telah terhubung dengan *mikrokontroler Arduino* uno R3. Dilengkapi dengan pompa air yang telah terhubung dengan *Relay* 5V. Apabila nilai kandungan tanah diatas nilai 900 maka kondisi tanah membutuhkan air sedangkan jika nilai dibawah nilai 400 maka kandungan tanah sedang lembab. Dengan smartphone yang telah terinstall aplikasi *Blynk* peneliti dapat menggunakan tombol *on/off* dalam aplikasi tersebut untuk menyiram tanaman.

Berdasar rumusan masalah dan penelitian sebelumnya, penulis akan merancang Aplikasi Pengontrol Sistem Penyiraman Tanaman Otomatis berbasis *Arduino* dan *Android*. Sensor *moisture soil* akan diletakkan di dalam tanah yang terhubung dengan *mikrokontroler* ESP32 berbasis *arduino*. Hasil pembacaan sensor kelembapan tanah yang diterima oleh *mikrokontroler* akan diupdate ke *realtime database* menggunakan platform *Firebase*. Kemudian data dari pembacaan sensor tersebut akan dipanggil ke dalam aplikasi *android* yang telah dirancang menggunakan bahasa pemrograman *dart* berbasis *Flutter*. Selain memberikan output data dari *realtime database*, aplikasi *android* tersebut juga dapat memberikan *input button* ke *realtime database* yang akan dibaca oleh *mikrokontroler* untuk diteruskan ke *relay* guna menyalakan dan mematikan pompa air. Perbedaan dengan penelitian yang telah dilakukan sebelumnya yaitu sistem penyiram tanaman ini dapat dilakukan pengontrolan kelembapan tanah dengan menyalakan tombol *on/off* secara langsung dari aplikasi *mobile* berbasis *android*. Serta

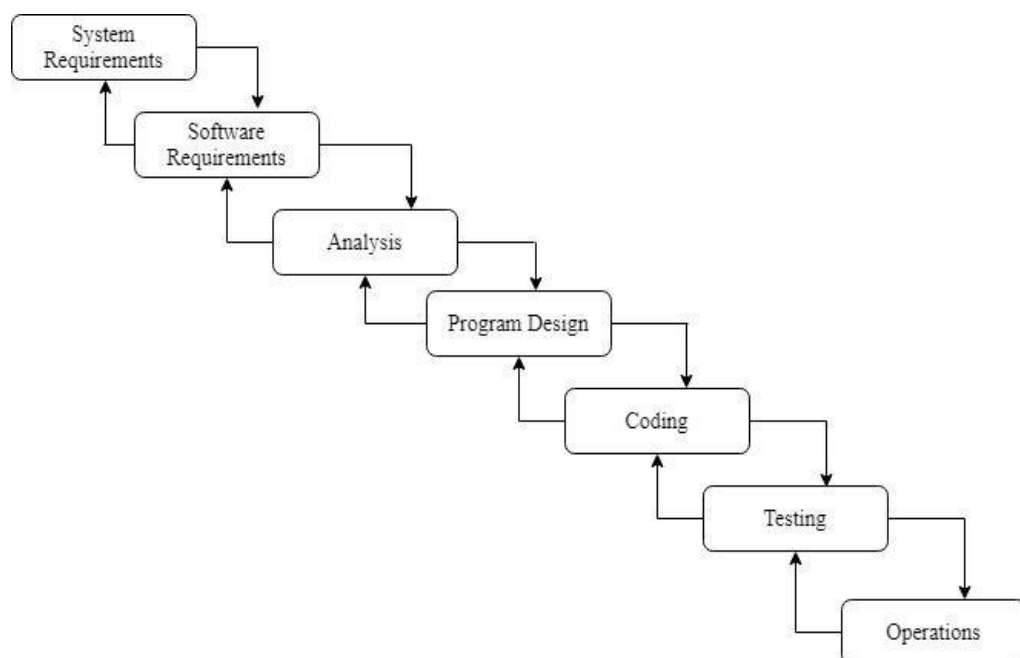
dapat memonitor pergerakan kelembapan tanah secara *realtime* yang berubah setiap saat. Sistem ini dapat mengontrol area dengan luas 1 m².

Dalam area 1 m² tersebut terdapat 6 tanaman dalam bentuk pot. Diharapkan sistem ini dapat mempermudah pemilik tanaman untuk mengontrol tanaman dari jarak jauh.

2 METODE

Metode pengembangan yang digunakan penulis pada sistem ini adalah SDLC (*System Development Life Cycle*) dengan model *waterfall*. Metode SDLC dengan model *waterfall* merupakan model terbaik dimana setiap tahap dimulai dan diakhiri sebelum tahap selanjutnya dikerjakan sehingga setiap tahapan tidak tumpang tindih dalam peroses pengerjaannya (Alshamrani & Bahattab, 2015).

Gambaran metode *System Development Life Cycle* dijelaskan pada gambar 1 sebagai berikut.



Gambar 1. Tahapan Metode *Waterfall* (Royce, 1987)

Tahapan metode *waterfall* dimulai dari tahap *system requirements* dengan menentukan kesepakatan untuk spesifikasi dari *software* dan *hardware* untuk sistem. Persyaratan akan *software* dan *hardware* yang telah didapat kemudian digunakan dalam menentukan *software requirements* bagi *user*. Kebutuhan akan sistem dan *software* kemudian dianalisa untuk selanjutnya dievaluasi dan dirancang arsitektur sistemnya

pada tahap *design*. Dengan arsitektur sistem yang telah terbentuk, pembuatan sistem dilakukan pada tahap *coding* untuk kemudian diujikan pada tahap *testing*. Sistem yang telah jadi kemudian dioperasikan pada tahap *operation* (Royce, 1987).

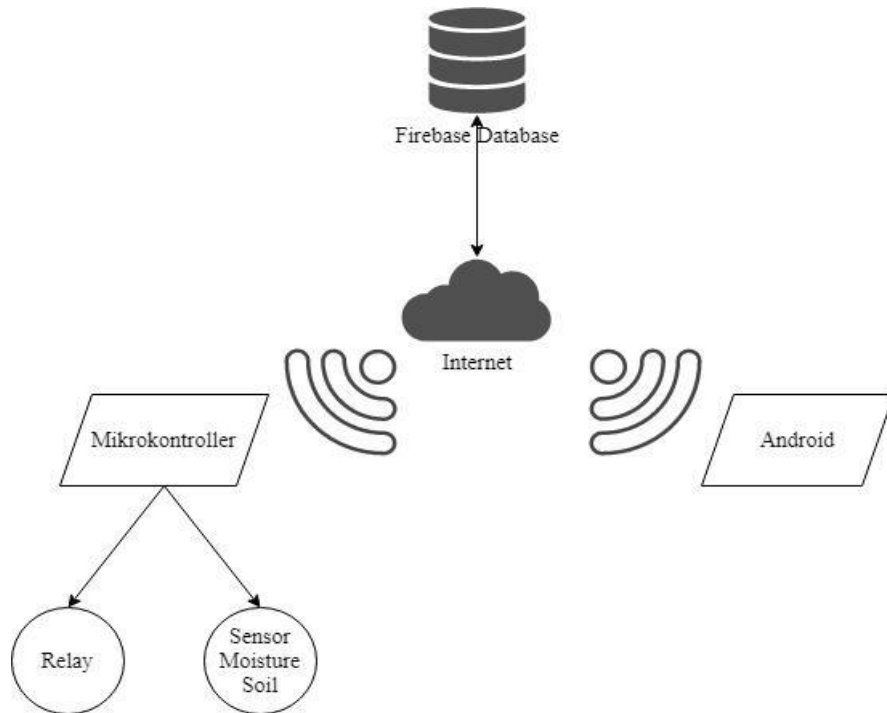
Aplikasi ini dirancang untuk *user android* yang nantinya memberikan kemudahan *user* dalam mengontrol dan menjalankan sistem. Proses-proses yang dapat dilakukan aplikasi adalah memberikan informasi data kelembapan tanah dari hasil pembacaan sensor dalam *mikrokontroler*, menampilkan data kelembapan tanah secara realtime dari setiap perubahan hasil pembacaan sensor, dan memberikan perintah kepada *mikrokontroler* untuk menyalakan dan mematikan pompa menggunakan tombol *on/off* pada aplikasi *android*.

Analisis sumber daya yang dibutuhkan untuk proses perancangan aplikasi dan pembuatan sistem merupakan tahapan dari persyaratan fungsional. Sumber daya tersebut terdiri dari *software* dan *hardware*.

Kebutuhan *software* yang diperlukan sistem yaitu aplikasi dalam perangkat *smartphone android* yang telah selesai dirancang. Sedangkan kebutuhan *hardware* berupa *Smartphone* berbasis *Android*, jaringan internet *wireless* menggunakan *WiFi* rumah, *Mikrokontroler* ESP32, Sensor *Moisture Soil*, *Relay*, dan pompa air.

Kebutuhan sistem perangkat lunak dan perangkat keras dialokasikan pada tahap perancangan sistem dengan membentuk arsitektur sistem secara keseluruhan. (Sasmito, 2017). Desain arsitektur sistem dan aplikasi ditampilkan dalam bentuk Arsitektur Sistem, Sketsa Lahan, *Use Case Diagram*, dan *Activity Diagram* untuk memudahkan pemahaman *user*.

Arsitektur sistem dari Pengontrol Penyiraman Tanaman ini terdiri dari *software* dan *hardware*. *Software* yang digunakan dalam sistem meliputi *software* yang terdapat di dalam *mikrokontroler*, *software* aplikasi dalam *android*, serta server *database* untuk penyimpanan *database*. Sedangkan *hardware* yang digunakan dalam sistem ini terdiri dari *mikrokontroler*, sensor moisture soil, *relay*, dan *smartphone android*. Gambaran arsitektur sistem dijelaskan dalam gambar 2.



Gambar 2. Arsitektur Sistem

Sistem ini menggunakan sensor moisture soil FC-28 yang terhubung secara langsung dengan *mikrokontroler* dan bekerja dengan membaca data kelembapan tanah untuk kemudian dikirim ke *mikrokontroler*. Sensor moisture soil FC-28 memiliki spesifikasi tegangan *input* 3.3 Volt atau 5 Volt dan tegangan *output* 0 – 4.2 Volt. Serta memiliki arus 35mA dan *value* range ADC sebesar 1024 bit (Husdi, 2018).

Mikrokontroler yang digunakan adalah ESP32 berbasis *Arduino* dengan spesifikasi RAM 520K, Prosesor Tensilica L108 32 bit, Kecepatan Prosesor Dual 160MHz, Tegangan 3.3 Volt, dan dilengkapi WiFi yang tertanam dalam *mikrokontroler* (Kusumah & Pradana, 2019). *Mikrokontroler* diprogram dengan bahasa pemrograman C menggunakan *software Arduino IDE* (Husdi, 2018) untuk memberi perintah kepada sensor dan *relay*.

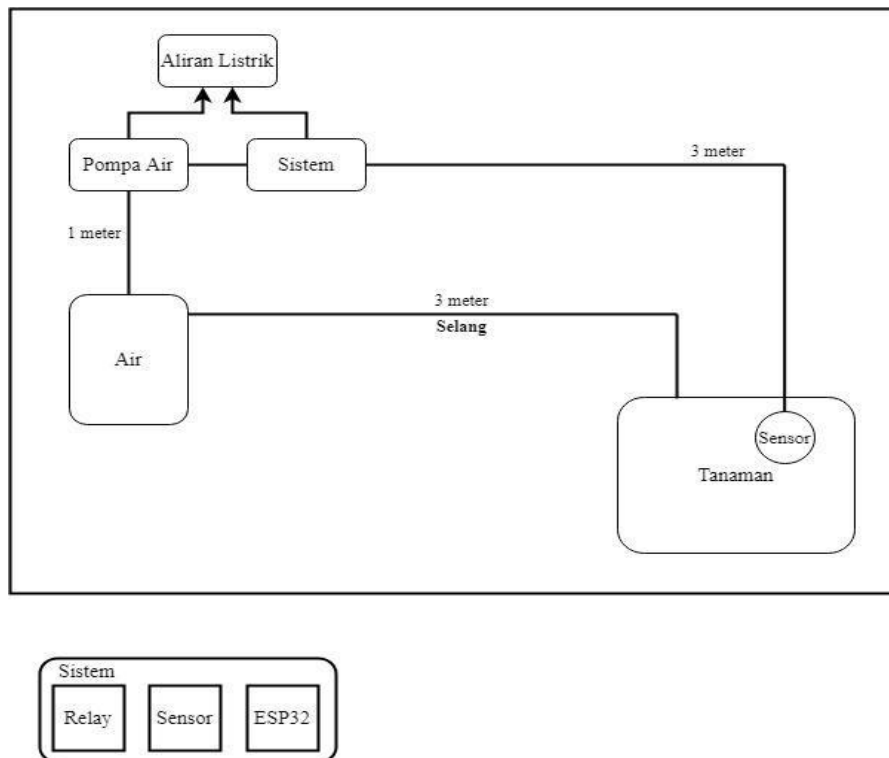
Perintah yang didapat *relay* dari *mikrokontroler* adalah untuk menyalakan atau mematikan pompa air. Pompa air terhubung dengan *relay* dan arus listrik secara langsung. *Relay* tersebut memiliki spesifikasi tegangan AC 125 V atau 250 V, Arus 10 A, dan Daya 1250 W atau 2500 W serta tegangan DC 28 V atau 30 V, Arus 10 A, dan Daya 128 W atau 300 W (Nurkamiden et al., 2017).

Data hasil pembacaan sensor yang diterima oleh *mikrokontroler* kemudian akan diteruskan ke *Server Firebase*. *Firebase* merupakan platform milik Google yang

memberikan layanan servis dan alat yang dibutuhkan *web development* maupun *mobile development* untuk mengembangkan aplikasi (Rajappa et al., 2020). Data kelembapan tanah tersebut disimpan dalam platform *firebase* yang berupa *hosting cloud* atau yang biasa disebut *firebase realtime database* (Al- Kababji et al., 2019).

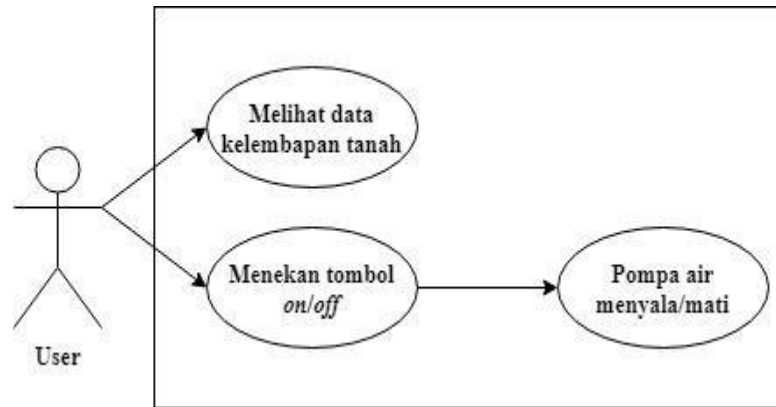
Selain menyimpan data kelembapan tanah, *firebase realtime database* juga menyimpan perintah menyalakan dan mematikan pompa air untuk diteruskan ke *mikrokontroler*. Data kelembapan tanah dan perintah *on/off* pompa air ditampilkan dalam aplikasi *android* yang telah di program menggunakan toolkit *flutter*. *Android*, *server firebase* dan *mikrokontroler* harus terhubung dengan internet untuk dapat menjalankan sistem dengan baik.

Berdasarkan arsitektur sistem pada gambar 2, dibuatlah sketsa implementasi lahan yang dapat dilihat pada gambar 3. Terdapat ruangan berisi *mikrokontroler* yang terhubung dengan arus listrik dan *relay* disampingnya. *Relay* terhubung langsung dengan pompa air untuk mengambil air pada galon di bawahnya pada jarak 1 m. Sedangkan sensor moisture soil tertanam dalam tanaman dengan luas 1 m² di luar ruangan.



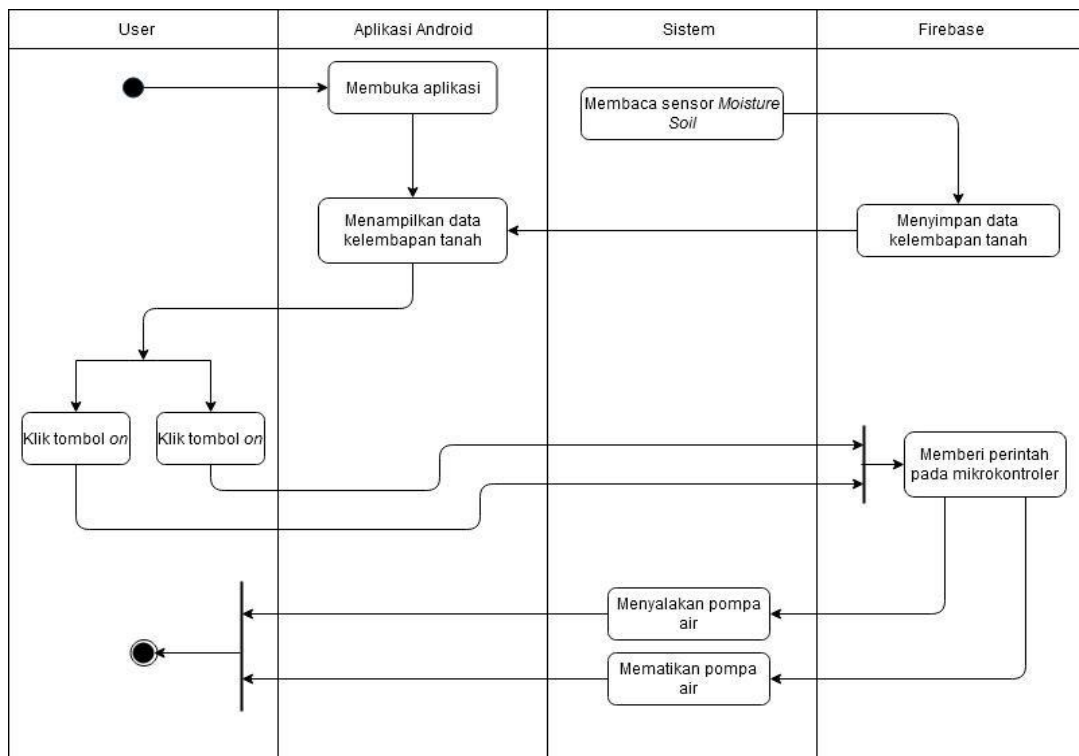
Gambar 3. Implementasi sketsa lahan

Interaksi antara *user* dengan sistem dijelaskan dalam *use case diagram*. Interaksi yang dijelaskan dalam *use case* berupa apa yang dilakukan *user* terhadap sistem dan apa yang disampaikan sistem kepada *user* (Muslihudin et al., 2018). *Use case diagram* sistem dapat dilihat pada gambar 3.



Gambar 4. *Use Case Diagram*

Interaksi *user* dengan sistem dijelaskan secara detail dalam *activity diagram* yang dapat dilihat pada gambar 5.

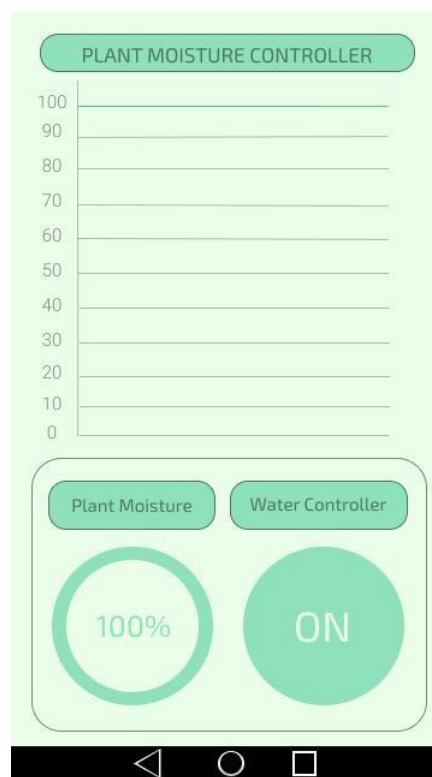


Gambar 5. *Activity Diagram*

Sistem membaca sensor kelembapan tanah secara berkala tanpa henti ketika mendapat jaringan internet. Data pembacaan sensor akan disimpan dalam *realtime*

firebase database. Saat aplikasi dalam *android* dijalankan, maka data yang tersimpan dalam *database* akan ditampilkan. Selain menampilkan data pembacaan sensor, aplikasi juga dapat memberi perintah *on/off* untuk menyalakan atau mematikan pompa air. Ketika tombol *on* dipilih, maka perintah akan diinput ke *database* guna diteruskan ke sistem untuk menyalakan pompa air. Begitu pula jika tombol *off* dipilih, maka perintah sebaliknya yang akan dikerjakan oleh sistem.

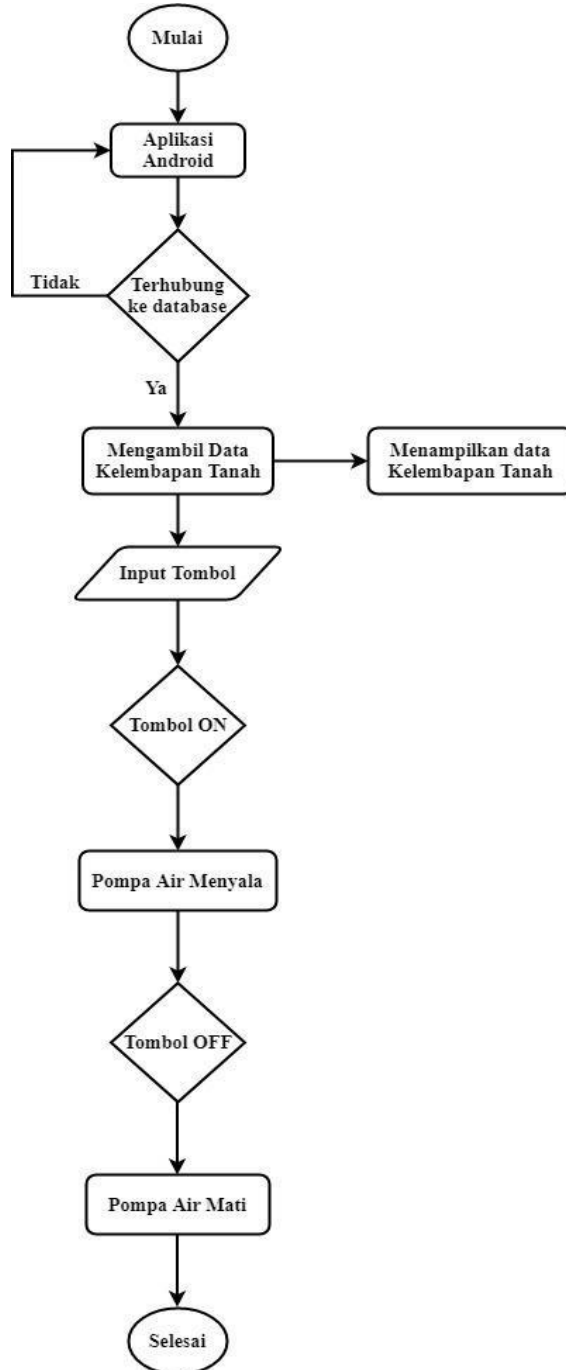
Desain *User Interface* dari aplikasi *android* dapat dilihat pada gambar 6, yang meliputi grafik data kelembapan tanah dalam satuan waktu, data kelembapan tanah dalam persen, serta *button ON/OFF* untuk menyalakan atau mematikan pompa.



Gambar 6. Desain *User Interface*

Setelah melewati tahap analisis kebutuhan dan desain, tahap selanjutnya adalah implementasi. Alur kerja dari aplikasi dan sistem secara keseluruhan ditampilkan dalam bentuk *flowchart*. Dalam gambar 7 *flowchart* sistem dimulai dengan membuka aplikasi *android*. Apabila aplikasi telah terhubung dengan *database* dalam *server firebase*, maka aplikasi akan mengambil data dari *database* dan menampilkannya dalam halaman utama aplikasi. Namun ketika aplikasi gagal terhubung dengan *database* dalam *server firebase*, maka aplikasi akan memuat ulang atau *loading* ulang. Halaman utama aplikasi

selain menampilkan data dari *database* juga terdapat *input button*. *Input button* yang ditombol oleh *user* dalam kondisi *ON* akan membuat pompa air menyala. Selanjutnya *input button* dalam kondisi *OFF* akan membuat pompa air mati dan selesai.



Gambar 7. *Flowchart* alur kerja aplikasi dan sistem Perancangan sistem ini diawali dengan merancang aplikasi yang akan dibuat menggunakan *toolkit flutter* dan bahasa pemrograman dart dengan *software* aplikasi Visual Code.

Database yang telah dibuat di *server firebase* memiliki JSON file yang nantinya akan digunakan dalam proses pembuatan aplikasi. File JSON yang telah didownload diinput ke dalam folder aplikasi yang dibuat. Koneksi dengan *firebase* kemudian dibuat dengan fungsi `final dbRef = FirebaseDatabase.instance.reference();`. Fungsi tersebut nantinya akan digunakan untuk mengambil data dalam *firebase* dan menginput data boolean ke *firebase*.

Menggunakan fungsi yang telah dibuat, data dalam *firebase database* dipanggil dengan metode *stream* yang dideklarasikan `stream: dbRef.child("Data").onValue,`. Metode *stream* tidak lepas dari metode *builder*. Pada metode *builder*, *library* yang digunakan ialah *library snapshot* dengan fungsi untuk mengimplementasikan data dari JSON ke *class data*. Jika hasil *snapshot* ditemukan data dan data tersebut tidak *error* serta *value* dari data tidak kosong `if (snapshot.hasData && !snapshot.hasError && snapshot.data.snapshot.value != null)` maka data akan muncul dengan memanggil metode *return*.

Dengan fungsi yang sama, dibuatlah metode baru untuk memberikan input ke *firebase database*. Sebelum membuat metode baru tersebut, digunakan *framework* `setState(() {value = !value});` yang berfungsi mengubah objek yang akan diinput antara *true* atau *false*.

Selanjutnya dibuat metode yang mendeklarasikan fungsi dan *framework* `writeData(){dbRef.child("Water").set({"Status": !value});}` untuk memberi input ke *firebase* yang mana akan diterima oleh *mikrokontroler* dan perintah "Status": !value tersebut dikerjakan oleh *relay*.

Aplikasi tersebut kemudian dihubungkan dengan sistem yang dibuat menggunakan *software Arduino IDE* pada *mikrokontroler ESP32*. Sebelumnya *mikrokontroler* dihubungkan dengan *relay* dan sensor moisture soil menggunakan kabel *jumper*. *Relay* terhubung dengan *mikrokontroler* pada *pin out 27* menggunakan kabel *jumper* serta terhubung dengan pompa air yang langsung teraliri arus listrik. Sedangkan sensor moisture soil terhubung dengan *mikrokontroler* pada *pin out A0* menggunakan kabel *jumper*. Ketika lahan yang menjadi target memiliki jarak yang cukup jauh, maka kabel *jumper* pada sensor disambung menggunakan kabel LAN.

Rangkaian *hardware* yang telah terpasang selanjutnya diinisialisasi

menggunakan *software Arduino IDE* pada *mikrokontroler ESP32*. *Mikrokontroler* diinisialisasi dengan jaringan WiFi rumah dan *server firebase* serta dihubungkan dengan sensor moisture soil dan *relay*.

Inisialisasi dengan jaringan WiFi rumah dilakukan dengan memasukkan WiFi SSD dan password WiFi pada *software Arduino IDE*. Selanjutnya proses inisialisasi terhadap *server firebase* dengan memasukkan *host* dan *auth* yang terdapat dalam *server firebase* pada *project* masing-masing.

Setelah terinisialisasi dengan WiFi dan *firebase*, *mikrokontroler* melalui *pin outnya* terhubung dengan sensor moisture soil dan *relay* menggunakan kabel *jumper*. Dengan kabel *jumper relay* terhubung dengan *mikrokontroler* pada *pin 27* `const int relay = 27;` dan sensor *moisture soil* pada *pin A0* `const int moisturePin = A0;`. Data kelembapan tanah didefinisikan dengan `int moisturePercentage;` dan selanjutnya dipanggil untuk diolah menggunakan perhitungan rumus `moisturePercentage = (100.00 - ((analogRead(moisturePin) / 4095.00) * 100.00));`. Data hasil perhitungan akan diteruskan ke *server firebase* `Firestore.setDouble(firebaseData, "/Data/Moisture", moisturePercentage);` sesuai dengan nama *database* yang telah dibuat pada *project firebase*.

Apabila sensor memberikan data, maka berbanding terbalik dengan *relay* yang mendapatkan data untuk kemudian dikerjakan. Jika data yang di dapat *relay* berasal dari *firebase* berupa `Firestore.getBool(firebaseData, "Water/Status")` maka akan dibaca *relay* dengan tipe data Boolean `RelayStatus = firebaseData.boolData();`. *Relay* berada pada kondisi *active low*, sehingga jika data *true* maka *relay* akan menyala `if (RelayStatus`

`== true){digitalWrite(relay,LOW);} sedangkan jika data false maka relay akan mati else {digitalWrite(relay,HIGH);}.`

Mikrokontroler yang telah selesai diprogram kemudian dihubungkan dengan arus listrik secara langsung untuk selanjutnya sistem dapat bekerja bersamaan dengan aplikasi yang telah di rilis.

Setelah proses pembuatan sistem pada tahap implementasi dilakukan proses pengujian atau *testing*. Pengujian dilakukan setelah aplikasi selesai dibuat guna menemukan kesalahan dan menilai efisiensi serta keefektifan aplikasi. Selain itu pengujian juga dilakukan terhadap sensor dan *relay* yang digunakan.

Jenis pengujian yang digunakan pada aplikasi *android* yaitu *black box testing*. Pengujian *black box testing* berpusat pada spesifikasi fungsional perangkat lunak (T. Hidayat & Muttaqin, 2018). Sedangkan pengujian pada sensor dan *relay* dilakukan dengan melihat tingkat keberhasilan dan keakuratan perangkat keras terhadap input yang diberikan.

Pengujian pada perangkat keras dilakukan dengan menghubungkan ESP32 ke laptop. Sensor moisture soil yang terhubung dengan ESP32 akan memberikan hasil pembacaan yang akurat dimulai dari angka 0 ketika tidak terkena air, dan akan bertambah perlahan jika terkena air. Hasil tersebut akan terlihat dalam *Serial Monitor* pada *Arduino IDE*.

Setelah diuji melalui *Arduino IDE*, sensor kemudian diuji dengan *firebase* apakah nilai hasil pembacaan sensor dapat ditampilkan dalam *database*. Sedangkan pengujian pada *relay* dilakukan melalui *firebase database* yang telah terhubung dengan ESP32. Dalam *firebase* diberikan input terhadap *relay* untuk kemudian dikerjakan. Jika input yang diberikan *TRUE* maka lampu pada *relay* akan menyala dan *Serial Monitor* menampilkan nilai 1. Dan jika input yang diberikan *FALSE* maka lampu pada *relay* akan mati dan *Serial Monitor* menampilkan nilai 0.

Selanjutnya *relay* diuji dalam kondisi telah terhubung dengan pompa air. Pengujian ini dilakukan untuk mengetahui apakah input yang diberikan ke *relay* berpengaruh terhadap pompa air. Jika input yang diberikan melalui *firebase* berupa *TRUE* maka pompa air akan bekerja, dan jika input berupa *FALSE* maka pompa air akan berhenti bekerja.

Pengujian selanjutnya dilakukan pada aplikasi *android* dan sistem secara bersamaan. Hal ini dilakukan untuk mengetahui apakah aplikasi dan sistem dapat berjalan sesuai yang diharapkan. Aplikasi akan menerima hasil pembacaan sensor dari *firebase*. Nilai akan akurat jika aplikasi menampilkan data realtime yang selalu berubah sesuai dengan yang terdapat dalam *firebase*. Dalam aplikasi juga terdapat button yang jika ditekan *ON* mengartikan pompa air akan bekerja dan jika ditekan *OFF* pompa air akan berhenti bekerja.

Pengujian terakhir yaitu *usability testing* dengan menggunakan instrument penelitian SUPR-Q (*Standardized Universal Percentilr Rank Questionnarie*) sebagai tolak ukur bagaimana sebuah perangkat lunak dikatakan mudah saat digunakan

(Pramitasari & Nurgiyatna, 2019).

Tahap akhir dari metode *waterfall* adalah *maintenance* atau pemeliharaan. Pemeliharaan dilakukan pada sistem yang sudah jadi untuk mengatasi munculnya *bug* atau *error*. Pengecekan terhadap sistem dilakukan secara berkala pada pemeliharaan guna menjaga kualitas aplikasi tetap baik.

3 HASIL DAN PEMBAHASAN

Bagian ini membahas tentang implementasi aplikasi dan sistem serta pengujiannya. Saat aplikasi dibuka pertama kali akan muncul *splash screen* seperti pada gambar 8.



Gambar 8. Tampilan *Splash screen*

Splash screen ini berfungsi sebagai *loading* sebelum masuk ke halaman utama pada aplikasi. Tampilan dalam *splash screen* tersebut berupa logo pohon dengan tulisan “Plant” di atasnya. Logo tersebut melambangkan tumbuhan yang berkaitan dengan aplikasi ini.

Setelah melewati *splash screen* muncul halaman utama aplikasi seperti pada gambar 9.



Gambar 9. Halaman utama aplikasi

Dalam halaman utama aplikasi terdapat data kelembapan tanah berbentuk grafik dan dalam bentuk *progress control* dengan label “Plant Moisture”. Terdapat 3 warna indikator dalam *progress control* berupa warna merah, oranye, dan hijau sesuai dengan data yang direspon.

Tombol dalam halaman utama aplikasi juga akan berubah ketika ditekan. Jika tombol dalam keadaan *OFF* ditekan, maka tombol tersebut akan berubah tampilan menjadi tombol *ON* dan begitu juga sebaliknya.

Implementasi dari sistem adalah terhubungnya seluruh perangkat keras seperti *relay* dan sensor moisture soil ke ESP32, *relay* dengan pompa air serta sensor moisture soil dengan tanaman seperti pada gambar 10.



Gambar 10. Implementasi perangkat keras

Pengujian dilakukan sebanyak 6 tahap. Pada aplikasi pengujian dilakukan menggunakan *black box testing*. Sedangkan pengujian terhadap sensor dan *relay* masing-masing 2 tahap. Serta tahap pengujian terakhir yaitu pada keseluruhan sistem dan aplikasi.

Black box testing digunakan sebagai pengujian pada aplikasi. Berikut hasil dari *black box testing* pada tabel 1.

Tabel 1. *Black box testing*

Kelas Uji	Pengujian	Input	Output	Hasil
Mulai aplikasi	User membuka aplikasi	Menekan logo aplikasi	Menampilkan <i>splash screen</i>	Valid
Tampilan awal aplikasi	<i>Loading</i>	<i>Loading</i>	Menampilkan halaman utama aplikasi	Valid
Kelas Uji	Pengujian	Input	Output	Hasil

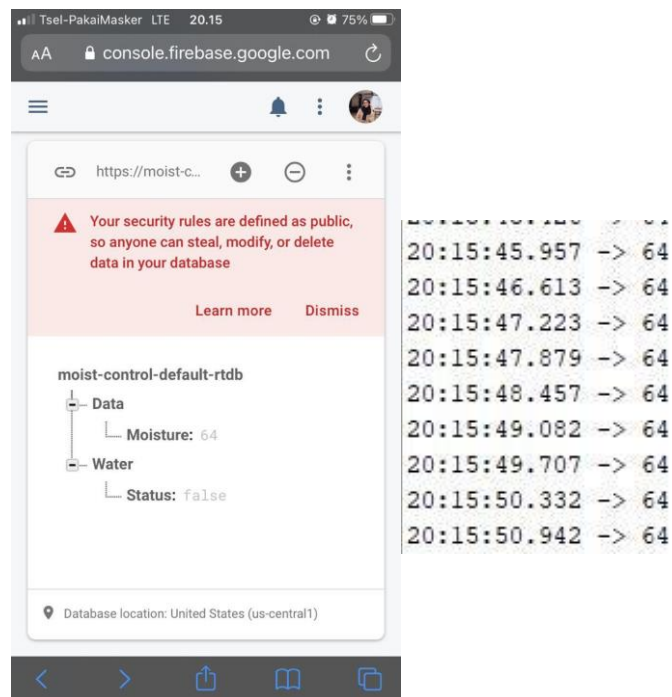
Halaman utama aplikasi	Melihat data kelembapan tanah	Tidak ada input dari <i>user</i>	Menampilkan data kelembapan tanah secara berkala dalam bentuk grafik dan indicator persen	Valid
	<i>User</i> menekan tombol pada <i>water controller</i>	Menekan tombol <i>ON</i>	Tombol men- <i>switch</i> menjadi tombol <i>OFF</i>	Valid
		Menekan tombol <i>OFF</i>	Tombol men- <i>switch</i> menjadi tombol <i>ON</i>	Valid
Keluar dari aplikasi	<i>User</i> menutup aplikasi	Menekan tombol keluar pada perangkat <i>mobile</i>	Menutup aplikasi	Valid

Pengujian tahap 1 dilakukan terhadap sensor moisture soil. Pengujian dilakukan untuk mengetahui apakah sensor berfungsi dengan baik dalam pembacaan kelembapan tanah. Pada tahap ini pengujian sensor dimonitor menggunakan *software Arduino IDE*. Pengujian dilakukan pada 04 Juni 2021 sebanyak 60 kali percobaan dengan pembagian waktu yang berbeda. Hasil dari pengujian sensor menampilkan nilai 40% - 45% pada pukul 07.24 – 07.25 WIB dengan kondisi tanah yang lembab, lalu nilai 55% - 54% pada pukul 12.09 –

12.10 WIB dengan kondisi tanah masih lembab, kemudian nilai 53% - 51% pada pukul 16.02 – 16.03 WIB dengan kondisi tanah yang masih tetap lembab, dan nilai 65% - 63% pada pukul 20.15 – 20.16 WIB dengan kondisi tanah yang basah. Satuan nilai kelembapan tanah dinyatakan persen dari hasil rata-rata kelembapan tanah harian (Karyati et al., 2018). Hasil pembacaan sensor tersebut sesuai dengan kondisi tanah dimana jika nilai yang ditampilkan kurang dari 20% maka kondisi tanah kering dan membutuhkan air, lalu jika nilai yang ditampilkan antara 20% dan 60% maka kondisi tanah lembab, sedangkan jika nilai yang ditampilkan lebih dari 60% maka kondisi tanah masih basah. Data tersebut diambil berdasarkan nilai kelembapan tanah mulai yang paling kering 0% dan paling banyak kandungan air 100% (Thomson et al., 2007).

Setelah dilakukan pengujian menggunakan *software Arduino IDE*, sensor diuji dengan membandingkan antara nilai yang muncul pada *serial monitor Arduino IDE* dan

nilai yang muncul pada *firebase*. Pengujian dilakukan secara bersamaan dengan pengujian sensor sebelumnya terhadap tanggal, waktu, dan banyaknya percobaan. Nilai yang muncul pada *firebase* selalu sama dengan nilai yang muncul pada serial monitor seperti pada gambar 11.



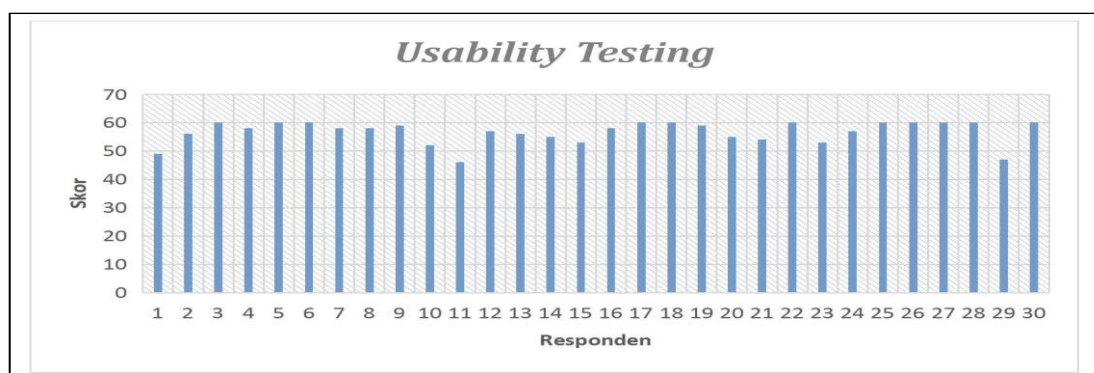
Gambar 11. Perbandingan nilai pada *firebase* dan serial monitor Selanjutnya pengujian terhadap *relay*. Pengujian dilakukan dengan

memberi input perintah berupa data *boolean* dari *firebase database*. Pengujian *relay* yang pertama dilihat dari respon indikator LED yang terdapat pada *relay*. Jika input yang diberikan *TRUE* maka indikator LED akan menyala, dan jika *FALSE* maka indikator LED akan mati. Pengujian dilakukan pada tanggal 01 Juni 2021 dan 03 Juni 2021 sebanyak 31 kali percobaan dengan rentang waktu yang berbeda. Hasil dari pengujian tersebut menyatakan bahwa ketika input *TRUE* diberikan maka LED akan menyala dan ketika *FALSE* maka LED akan mati dengan *delay* 0 detik atau tanpa *delay*.

Jika pengujian *relay* terhadap indikator LED telah selesai, selanjutnya dilakukan pengujian *relay* terhadap pompa air. Seperti dengan pengujian *relay* sebelumnya, input diberikan dari *firebase* berupa kondisi *TRUE* dan *FALSE*. Apabila input *TRUE* maka pompa air akan menyala dan bekerja. Sedangkan apabila input *FALSE* maka pompa air akan mati dan berhenti bekerja. Pengujian dilakukan secara bersamaan dengan pengujian *relay* sebelumnya terhadap tanggal, waktu, dan banyaknya percobaan. Hasil pengujian menyatakan bahwa ketika diberi input *TRUE* maka pompa air menyala dan bekerja, sedangkan jika diberi input *FALSE* maka pompa air akan mati dan berhenti bekerja dengan *delay* 0 detik atau tanpa *delay*.

Tahap selanjutnya adalah menguji sistem dan aplikasi *android* secara keseluruhan. Hasil pengujian ini menentukan apakah sistem dan aplikasi *android* telah bekerja sesuai dengan yang diharapkan. Pengujian dilakukan pada tanggal 04 Juni 2021 sebanyak 60 kali percobaan dengan pembagian waktu yang berbeda. Hasil dari percobaan tersebut menyatakan bahwa nilai yang ditampilkan pada aplikasi *android* bernilai sama dengan nilai yang ditampilkan dalam *firebase*. Dan ketika tombol *ON* ditombol maka *relay* akan menyala dan menyiram, sedangkan jika tombol *OFF* ditombol maka *relay* akan mati dan berhenti menyiram dengan *delay* 0 detik atau tanpa *delay*.

Setelah dilakukan beberapa pengujian sistem, selanjutnya dilakukan *usability testing* dengan menyebarkan kuisioner terhadap 30 responden dari minimal 20 responden (Kaikkonen et al., 2005). Skor dari setiap responden dapat dilihat pada gambar 12.



Gambar 12. Grafik *usability testing*

Skor setiap responden tersebut dihitung dengan rumus dalam SUPR-Q (Fay, 1967) pada persamaan (1) untuk mendapatkan Total Skor.

$$\begin{aligned}\text{Total Skor} &= \text{Total Nilai} \times \text{Total Pertanyaan} \\ &= 5 \times 12 \\ &= 60\end{aligned}\quad (1)$$

Keterangan :

Total Nilai : Banyaknya pilihan nilai
Total Pertanyaan : Banyaknya pertanyaan yang diajukan

Total Skor yang didapat kemudian digunakan untuk menghitung presentase menggunakan persamaan (2).

$$\text{Presentase (\%)} = \frac{\text{Skor}}{\text{Total Skor}} \times 100 \quad (2)$$

Keterangan :

Skor : Jumlah skor dari masing-masing responden
Total Skor : Perkalian antara total nilai dan total pertanyaan yang diajukan

Dari survey tersebut didapat perolehan hasil perhitungan pada kategori “Baik” sebesar 10% dan kategori “Sangat Baik 90%” dengan proses perhitungan berdasarkan SUPR-Q. Rata-rata presentase yang diperoleh dari perhitungan berkeseluruhan ialah 94,44% dengan kategori “Sangat Baik”.

4 PENUTUP

Aplikasi pengontrol sistem penyriman tanaman ini telah selesai dirancang. Dengan dikombinasikan bersama perangkat keras ESP32 yang saling terhubung dengan sensor moisture soil dan *relay*. Aplikasi ini memiliki fitur untuk melihat data kelembapan tanaman secara realtime dari sensor dan tombol untuk mengontrol aktifitas pompa air yang terhubung ke *relay*. Berdasarkan hasil pengujian *black box* fitur dalam aplikasi telah berjalan dengan semestinya. Dan dari *usability testing* dihasilkan rata-rata presentase sebesar 94,44% dengan kategori “Sangat Baik” yang artinya para responden setuju aplikasi ini membantu pekerjaan mereka.

DAFTAR PUSTAKA

Al-Kababji, A., Shidqi, L., Boukhenoufa, I., Amira, A., Bensaali, F., Gastli, M. S., Jarouf, A., Aboueata, W., & Abdalla, A. (2019). Iot-based fall and ecg monitoring system: Wireless communication system based firebase realtime

database. *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI* 2019, 1480–1485. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00267>

- Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, 12(1), 106–111. https://www.academia.edu/10793943/A_Comparison_Between_Three_SDL_C_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model
- Azzaky, N., & Widianoro, A. (2021). Alat Penyiram Tanaman Otomatis Berbasis Arduino Menggunakan Internet Of Things (IOT). *Jurnal Elektronika, Listrik, Telekomunikasi, Komputer, Informatika, Sistem Kontrol (J-Eltrik)*, 2(2), 86–91. <https://doi.org/10.30649/j-eltrik.v2i2.48>
- Fay, D. L. (1967). Usability Evaluation. *Angewandte Chemie International Edition*, 6(11), 951–952.
- Hidayat, F. (2019). Purwarupa Alat Penyiram Tanaman Otomatis menggunakan Sensor Kelembapan Tanah dengan Notifikasi Whatsapp. *Prosiding Semnastek*, iv, 1–2.
- Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jurnal Teknik Informatika UNIS JUTIS*, 6(1), 2252–5351. www.ccsenet.org/cis
- Husdi, H. (2018). Monitoring Kelembapan Tanah Pertanian Menggunakan Soil Moisture Sensor Fc-28 Dan Arduino Uno. *ILKOM Jurnal Ilmiah*, 10(2), 237–243. <https://doi.org/10.33096/ilkom.v10i2.315.237-243>
- Kaikkonen, A., Kekäläinen, A., Cankar, M., Kallio, T., & Kankainen, A. (2005). Usability testing of mobile applications: a comparison between laboratory and field testing. *Journal of Usability Studies*, 1(1), 4–16.
- Karyati, K., Putri, R. O., & Syafrudin, M. (2018). Suhu Dan Kelembapan Tanah Pada Lahan Revegetasi Pasca Tambang Di Pt Adimitra Baratama Nusantara, Provinsi Kalimantan Timur. *Agrifor*, 17(1), 103. <https://doi.org/10.31293/af.v17i1.3280>
- Kusumah, H., & Pradana, R. A. (2019). Penerapan Trainer Interfacing Mikrokontroler Dan Internet of Things Berbasis Esp32 Pada Mata Kuliah Interfacing. *Journal CERITA*, 5(2), 12–134. <https://doi.org/10.33050/cerita.v5i2.237>
- Muslihudin, M., Renvilia, W., Taufiq, Andoyo, A., & Susanto, F. (2018). Implementasi Aplikasi Rumah Pintar Berbasis Android Dengan Arduino Microcontroller. *Jurnal Keteknikan Dan Sains*, 1(1), 23–31.

- Nurkamiden, M. R., Najoan, M. E. I., & Putro, M. D. (2017). Rancang Bangun Sistem Pengendalian Perangkat Listrik Berbasis Web Server Menggunakan Mini PC Raspberry Pi Studi Kasus Gedung Fakultas Teknik Universitas Sam Ratulangi. *Jurnal Teknik Informatika*, 11(1).
<https://doi.org/10.35793/jti.11.1.2017.15980>
- Pambudi, A. S., Andryana, S., & Gunaryati, A. (2020). Rancang Bangun Penyiraman Tanaman Pintar Menggunakan Smartphone dan Mikrokontroler Arduino Berbasis Internet of Thing. *Jurnal Media Informatika Budidarma*, 4(2), 250.
<https://doi.org/10.30865/mib.v4i2.1913>
- Pramitasari, B., & Nurgiyatna, N. (2019). Sistem Informasi Unit Kegiatan Mahasiswa Universitas Muhammadiyah Surakarta Berbasis Web. *Emitor: Jurnal Teknik Elektro*, 19(2), 5965.
<https://doi.org/10.23917/emitor.v19i2.7998>
- Rajappa, A., Upadhyay, A., Sai Sabitha, A., Bansal, A., White, B., & Cottrell, L. (2020). Implementation of PingER on android mobile devices using firebase. *Proceedings of the Confluence 2020 - 10th International Conference on Cloud Computing, Data Science and Engineering*, 6, 698–703.
<https://doi.org/10.1109/Confluence47617.2020.9058306>
- Ridarmin, R., & Pertiwi, Z. P. (2018). *Prototype Penyiram Tanaman Hias Dengan Soil Moisture Sensor Berbasis Arduino*. 10(1), 7.
<https://doi.org/10.36723/juri.v10i1.54>
- Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. *Ideas That Created the Future, August*, 321–332.
<https://doi.org/10.7551/mitpress/12274.003.0035>
- Sasmito, G. W. (2017). *Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal*. 2(1), 6–12.
- Satria, D. (2019). *Desain Prototype Penyiraman Perkebunan Berbasis Android*). 3(2), 250–256.
- Sreeram, K., Kumar, R. S., Bhagavath, S. V., Muthumeenakshi, K., & Radha, S. (2018). Smart farming - A prototype for field monitoring and automation in agriculture. *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017, 2018- Janua*, 2189–2193.
<https://doi.org/10.1109/WiSPNET.2017.8300148>
- Thomson, M. J., Septiningsih, E. M., Suwardjo, F., Santoso, T. J., Silitonga, T. S., & McCouch, S. R. (2007). Genetic diversity analysis of traditional and improved Indonesian rice (*Oryza sativa* L.) germplasm using microsatellite markers. *Theoretical and Applied Genetics*, 114(3), 559–568.
<https://doi.org/10.1007/s00122-006-0457-1>
- Warangkiran, I., Kaunang, I. S. T. G., Lumenta, A. S. M., & St, A. M. R. (2014). *Perancangan Kendali Lampu Berbasis Android. 1*.